

AL66 FG ARM Pulse / mod. ccTalk COIN ACCEPTOR

Operator's Manual

Rev. 2.01



User Manual

CE



A.u.S. SPIELGERÄTE GMBH

SCHEYDGASSE 48, AT 1210 WIEN

+43 (0) 1 271 66 00 65 FAX +43 (0) 1 271 66 00 75

www.aus.at ✉ verkauf@aus.at

CONTENTS

| | |
|--|----|
| 1. Package content | 5 |
| 2. Product identification..... | 5 |
| 3. Product description | 5 |
| 4. Technical specifications | 6 |
| 5. Installation | 7 |
| 6. ccTalk Communication Protocol | 12 |
| 7. Driving SPI peripheral devices..... | 28 |
| 8. Maintenance | 31 |
| 9. Disposal of the product..... | 32 |
| 10. Warranty | 32 |
| 11. Customer Service | 32 |

| STORICO REVISIONI | | | |
|-------------------|----------|--|-------------|
| Revisione n° | Data | Modifica | Note |
| Creazione v. 2.00 | 13.02.20 | Creazione versione con Microprocessore ARM 3 | Fw: u3.1A + |
| Versione v. 2.01 | 18.09.20 | Inserito separatore iS4 | |

NOTICE

This manual has been drafted with the utmost care. Nevertheless, it is not possible to guarantee at all times the absolute correspondence of the descriptions contained therein with the actual characteristics of the product. Alberici S.p.A. declines any and all responsibility towards the User with reference to damages, losses, or claims of third parties, resulting from the use of the product or caused by incorrect interpretations of this manual. Alberici S.p.A. reserves the right to modify, without prior notice and in any way, any part of this manual and the technical specifications of this product, as part of the continuous pursuit of improvement of its products.

Dear Customer,

we would like to thank you and congratulate you on your choice of the AL66 FG ARM Pulse / ccTalk electronic coin selector. We are confident that you will appreciate the quality and performance.

Please read this manual carefully to get the most from this product.

1. Package content

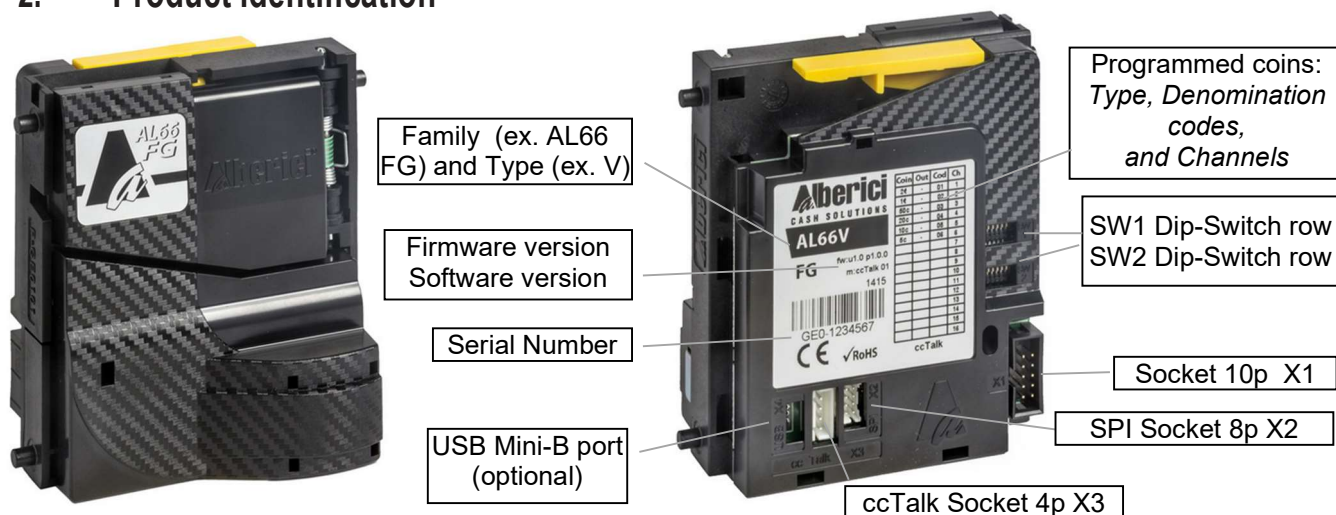
The package contains:

1. The AL66 FG ARM Pulse / cctalk coin acceptor (type V, or type S, or type I)
2. The manual of the product (this manual), or the Data sheet

This product has been packed with the utmost care. If you receive it damaged or incomplete, please notify immediately your findings to the Carrier, and collect it only under conditional signature.

The packing material, wooden pallets included, must be disposed off according to your Country laws and regulations. The product itself, once dismissed, must be disposed off according to the same laws and regulations.

2. Product identification



3. Product description

The architecture of the AL66 FG coin acceptor has been designed specifically for the ccTalk protocol, on which solid basis the Pulse mode operates as well, in any of the possible programmable configurations. The ccTalk signals sent to the machine CPU are channeled through the classic 4-pole TTL connector (ccTalk protocol: dip-switch 1 in SW1 = ON); the Pulse signals go through the 10p socket (Pulse communication: dip-switch 1 in SW1 = OFF).

The AL66 FG is equipped with the latest generation 32-bit ARM microprocessor, with 128 KB flash memory, "automotive" model, whose performance is stable even in difficult environmental conditions. Thanks to its 6 sensors, 2 pairs of optical sensors and innovative detection techniques, it generates 13 dynamic recognition parameters that ensure excellent performance in identifying coins and discriminating counterfeits. Anti-fraud protection is guaranteed by 4 directional optical pairs and mechanical devices (wire cutter and wire stripper) which protect against "fishing", "pendulum", and other types of tampering. It is equipped with mechanical and electronic anti-cheat devices and controls, against the most widespread and recent scam techniques. The construction, with non-deformable HQ polymer resins, ensures strength and durability, helping to ensure constant reading conditions regardless of the environment in which the coin mechanism is working. Using the Dip-Switches of bank SW1, coins to be accepted can be disabled or enabled. The acceptance tolerance can be changed dip-switch 6 of SW2 row. Again using the dip-switches of the SW2 counter it is possible to increase or decrease the credit value (service activation price) when the coin mechanism is configured as Pulse Totalizer. In addition, the signal output can be moved from the standard pre-assigned output (OUT 7) to OUT 3.

4. Technical Specifications

- > 32-bit top-performing ARM microcontroller with 36 KB Flash-memory, highly immune to magnetic interference and to environmental conditions.
- > Six magnetic sensors and one optic detector combine to guarantee the most accurate capacity of discrimination. Data-digitizer interface copes at best with bi-metal and magnetized coins.
- > One wire-shearer (and one wire-tearer in version V) provide mechanical

The AL66 FG coin acceptor can operate by Pulse mode or by modifiable ccTalk protocol.

It is available in the following versions:

V = rejected coin sorted from bottom rear slot, accepted coin sorted from bottom front slot
I = rejected coin sorted from bottom front slot, accepted coin sorted from bottom rear slot

S = rejected coin sorted from slot low front end, dropping slot at upper front end



Technical specs

Mechanical features

| | |
|--------|------------------|
| Format | 3½" standard |
| Size | 88 x 102 x 52 mm |
| Weight | 212 g |

Electrical features

| | |
|----------------------------|----------------------------|
| Supply voltage min. | 12 V DC (min. 10 V DC) |
| Supply voltage max. | 24 V DC (max. 26 V DC) |
| Power consumption: | |
| in acceptance | max. 350 mA(30 ms)/100 mA |
| in reading | ≤30mA |
| in Standby | ≤25mA |
| Output type | Open collector Darlington |
| Saturation output voltage | ≤1 V |
| Output voltage max. | 50 V (Active Low) |
| Output current max. | 250 mA |
| Input trigger voltage min. | 3 Vdc/24 Vdc (Active High) |
| Input voltage max | 50 V |
| Input impedance | ≈55 kfi |

Coin acceptance

| | |
|----------------------|---|
| Coin channels number | 16 |
| Coin diametre min. | 16 mm |
| Coin diametre max. | 32 mm |
| Coin thickness | 1 to 3,4 mm |
| Speed | 3 coins/sec. (V) – 4 coins/sec. (S, K, I) |

Communication modes

| | |
|-------------------|-------------------------------|
| Pulse | switching by Dip-Switch or by |
| Modifiable ccTalk | programming software |

Timing data

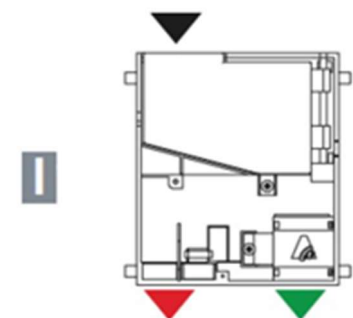
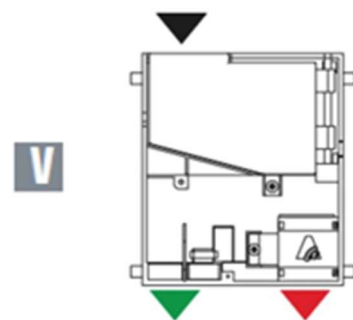
| | |
|-----------------------------|----------|
| Power-up recovery time | ≤ 200 ms |
| Wake-up recovery time | ≤ 50 ms |
| Pulse or time out tolerance | ± 2% |

Ambient conditions

| | |
|-----------------------------|--|
| Working ambient temperature | 0°C to 60°C |
| Storage temperature | - 30°C to 70°C |
| Humidity | up to 75% (<i>non condensing</i>) for standard up to 95% for tropicalized version |

EMC performance

This product is compliant with EN55014-1 and EN55014-2 test specification



INGRESSO MONETA COIN IN

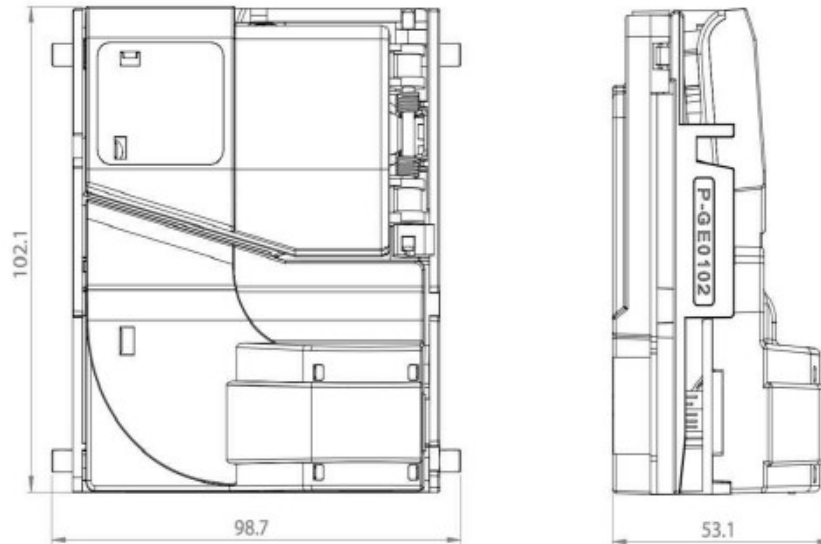
MONETA VALIDA VALID COIN

MONETA RIFIUTATA REJECTED COIN

5. Installation

DIMENSIONS

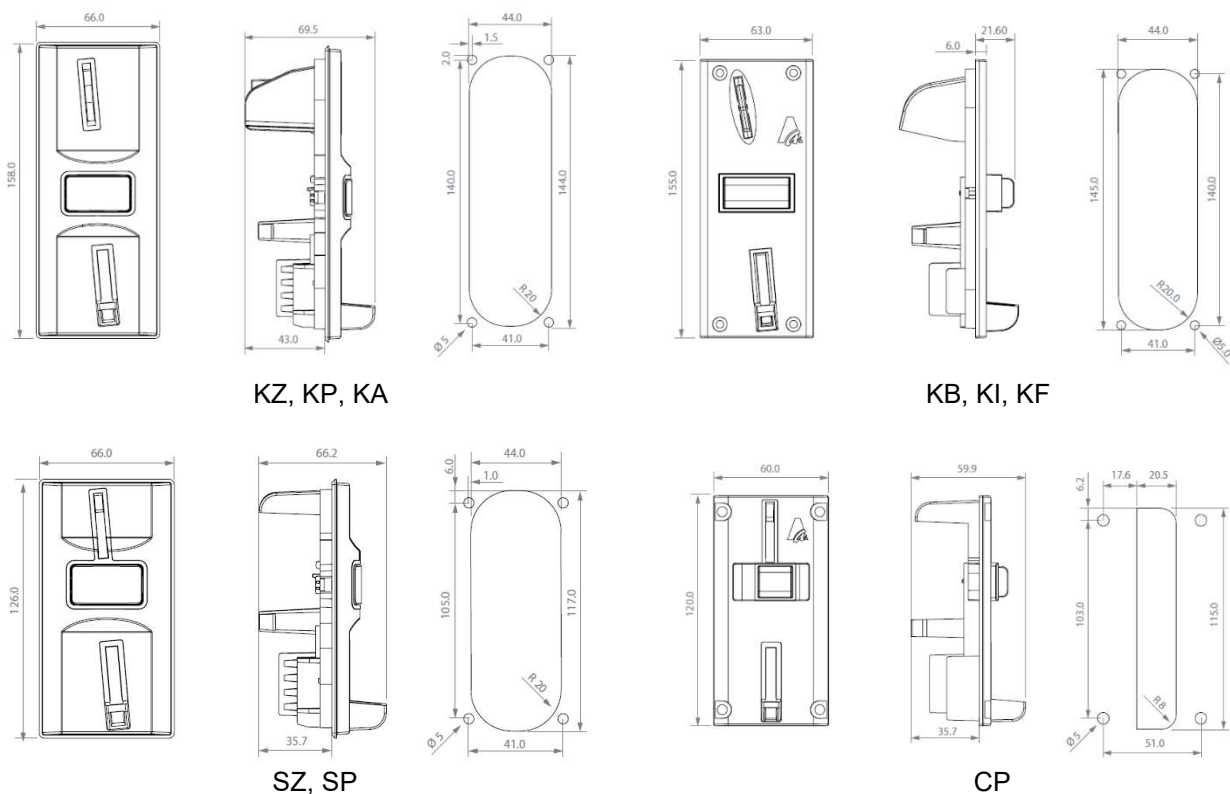
The AL66 FG is a 3,5" coin acceptor. It is interchangeable with the coin acceptors of the same type (compatible entries and exits of the coins) available on the market.

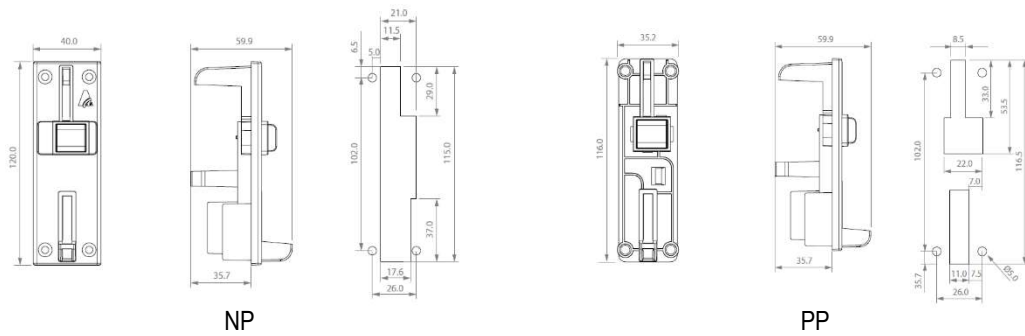


PAY ATTENTION! The coin selector must be installed 90°-95° with respect to level plane. Because of the built-in advanced security system, it is essential that the whole coin path gets not hindered. Alberici is not responsible for any malfunctioning due to lack of compliance with such recommendations.

MOUNTING of the "S-TYPE COIN ACCEPTOR (FACE PLATES)

Various models of front entries are available, to whose bracket the coin mechanism gets firmly hooked. The KZ, KP, KA, SZ, and SP face plates are designed to be mounted outside the installation panel. The KB, KI, KF, CP and NP plates can be fitted both externally and internally. The PP introduction is specially designed to be mounted inside, so that the slots are flush with the external profile of the panel.





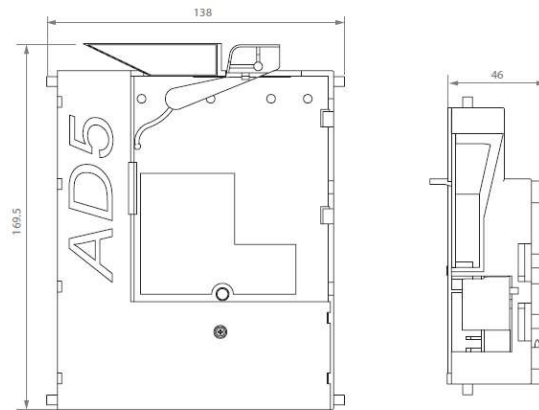
NP

PP

The sorter can be mounted directly at the bottom of the “S”-type coin acceptor, by the dedicated AA-0703 bracket.

MOUNTING of the “S”-TYPE COIN ACCEPTOR in the 5” ADAPTOR

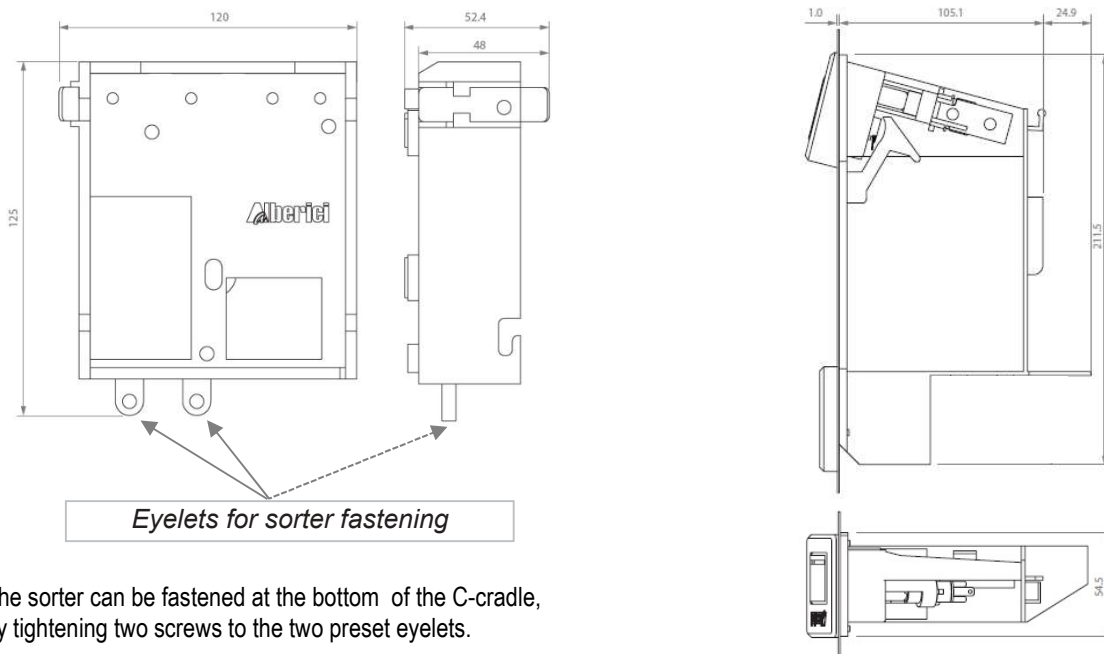
When incorporated into the 5” adaptor (SG-5000), and by adding the upper reject lever and the relevant torque spring, the 3,5” coin acceptor can be used instead of 5” coin acceptors having the same communication protocol.



The 5” casing is available also with built-in sorter, and/or with electronic rejecting device for remote installation of the coin acceptor.

MOUNTING of the “V”-TYPE COIN ACCEPTOR in C-CRADLE or in “SPANISH” CRADLE

The “V”-type (or “video”) coin acceptor must be housed into the “C”-cradle (SG-7000) or into “Spanish” type cradles (SG-3001 / SG-4001). The preferred drop-in coin slot can be fitted onto the upper side of the C-cradle, or else the electronic rejecting device when the coin acceptor must be installed far from the entry slot.



Eyelets for sorter fastening

The sorter can be fastened at the bottom of the C-cradle, by tightening two screws to the two preset eyelets.

Connections

The selector connects to other peripherals and to the main board by the following sockets:

X1. Power supply and standard Pulse interface.

Connector X1 is an IDC 10-p socket (pin-out shown in grid here aside). It consists of: 2 power supply pins (*pin 1 common/ground & pin 2 positive*), 1 input (*pin6, normally used to inhibit acceptance of coins*), and 7 “open collector” outputs (*pins 3,4,5,7,8,9 & 10*), one of which (pin 5) is normally used for the multi-pulse totalizer (“open collector” output), or else as additional input (i.e. for credit request).



| nr. | Description |
|-----|--------------------------|
| 1 | Gnd |
| 2 | +12-24 Vdc |
| 3 | Out 5 / sorter coil B |
| 4 | Out 6 / sorter coil A |
| 5 | Out 7 (totalizer) / In 2 |
| 6 | In 1 (inhibit) |
| 7 | Out 1 |
| 8 | Out 2 |
| 9 | Out 3 |
| 10 | Out 4 / sorter coil C |

OUTPUTSACTIVESTATE=LOW (reversible)
INPUTSACTIVESTATE=HIGH(reversible)

Upon request, the acceptor can be prepared for reverse supply polarity, so as to emulate Spanish-style coin acceptors.

X2. SPI Display or intelligent peripherals.

Socket X2 (8 pins) is meant for display connection. Displays with SPI or I²C bus protocols are supported (see DISPLAY AD01 SPI SERIAL PROTOCOL at page 20).



The acceptor must be preset by Alberici to host the display (AA-0410-0900). Display functions can then be modified by the ‘AlbericiCoinSelector’ programming software.

X3. CCTALK. The X3 4-p socket is for serial **ccTalk**® communication with the main board. The protocol is preset for “slave mode” operation, and is described in Chapter 6. This socket can be used in Pulse and standard cctalk selectors to modify the programmed configuration of the selector, by PC and Alberici free software.



| nr. | Description |
|-----|-------------|
| 1 | Data |
| 2 | Gnd |
| 3 | NC |
| 4 | +12 Vdc |

X4. USB Mini-B (option). The USB port allows to drive the acceptor directly the USB port of a PC Mother Board. The USB port provides only the communication channel: the communication itself will be run by serial **ccTalk**® protocol, that must therefore be implemented in the Host software (see Chapter 6).

To provide USB operation, the acceptor must be +12V powered through pins 2 and 4 in socket X3. This socket will be enabled only after installation of the USB drivers on the Host (See ‘USB installation notice’ below).



USB pre-activation must be requested at the time of order.

Please be aware that USB coinmechs do not allow Operation in Pulse mode.

USB installation notice

- Install the drivers in the Host using the relevant executable file (i.e.: for Windows 7 64bit: “VCP_V1.5.0_Setup_W7 x64_64bits.exe”), available in the Alberici web site (FG coin acceptor page). Win 10 includes such drivers by default.
- Only after the drivers (“STMicroelectronics Virtual COM Port”) are installed, connect the coin acceptor through the USB;
- Windows OS will automatically start the installation of the driver;
- At the end of the installation process, a dialogue window will confirm “STMicroelectronics Virtual COM Port (COMxx)”.
- It is now possible to connect +12V power to the pins 2 (GND) and 4 (+) of the X3 socket.

PAY ATTENTION! Using a standard ccTalk interface circuit, the master (i.e. host mother board) receives the echo back from the interface, and NOT from the slave (i.e. coin selector).

However, USB port is NOT a standard ccTalk interface circuit: no TX echo will be sent to the master!

Therefore the master must be programmed so as not to expect to receive the echo, or communication will end.

DISABLE DENOMINATIONS BY THE SW1 ROW

EACH PROGRAMMED CHANNEL CONTAINS ONE DENOMINATION. TO DISABLE ACCEPTANCE OF ANY COINS, MOVE TO OFF POSITION THE DIP-SWITCH CORRESPONDING TO THE CHANNEL IN WHICH THE COIN IS PROGRAMMED (see grid below). THEN SWITCH OFF AND ON AGAIN.

| SW1 Dip-Switch Row | DS1 | DS2 | DS3 | DS4 | DS5 | DS6 |
|--------------------|-------------------------|-------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| ON | Enable CH1 (ex. 2€) | Enable CH2 (ex. 1€) | Enable CH3 (ex. 0,50€) | Enable CH4 (ex. 0,20€) | Enable CH5 (ex. 0,10€) | Enable CH6 (ex. 0,05€) |
| OFF | Disable CH1 (ex. 2€) | Disable CH2 (ex. 1€) | Disable CH3 (ex. 0,50€) | Disable CH4 (ex. 0,20€) | Disable CH5 (ex. 0,10€) | Disable CH6 (ex. 0,05€) |

SETTINGS BY THE SW2 DIP-SWITCH ROW

| SW2 Dip-Switch Row | DS2 OFF | DS2 ON | DS4 OFF | DS4 ON | |
|--------------------|---------------------|---------|---------|--------|---|
| DS1 OFF | PULSE | MDB (*) | | | |
| DS1 ON | CCTALK | SAS (*) | | | |
| DS3 OFF | | | | | Multi-pulse totalizer, OUTPUT as per programmed setting |
| DS3 ON | | | | | Multi-pulse totalizer on OUTPUT 3 |
| DS5 OFF | Modify credit value | | x 1 | x 4 | |
| DS5 ON | | | x 2 | : 10 | |
| DS6 OFF | | | | | STANDARD Discrimination (L / SLC) |
| DS6 ON | | | | | HIGH Discrimination (H / SLC) |


(*) Not available in this version

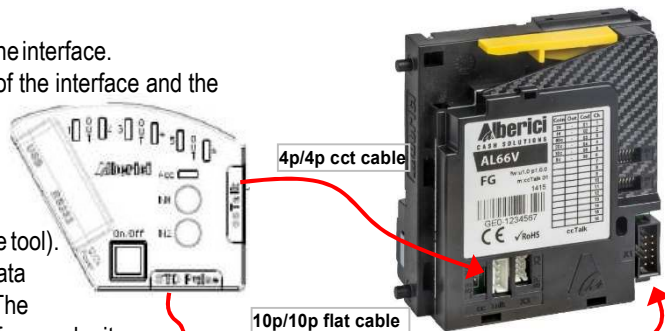
| HOW TO TEACH (NEW) COINS BY THE DIP-SWITCHES | CANCEL THE LAST SELF-TEACHED CONFIGURATION |
|--|---|
| <ol style="list-style-type: none"> 1) Take note of the positions of the dip-switches. 2) Power off the coin acceptor. 3) Set all the dip-switches in SW1 row to OFF. 4) Move the dip-switch 2 of the SW2 row to ON. 5) Power up the coin acceptor: start of the auto-programming mode is confirmed by single coil activation. 6) In SW1 row, move to ON the dip-switch corresponding to the channel in which the coins will be taught. 7) Drop 10 such coins into the acceptor: double coil activation will confirm that the coins have been programmed. 8) Power off the coin acceptor and set the dip-switches according to the desired operation. | <ol style="list-style-type: none"> 1) Take note of the positions of the dip-switches. 2) Power off the coin acceptor. 3) Set all the dip-switches in SW1 row to OFF. 4) Move the dip-switch 2 of the SW2 row to ON. 5) Set to ON both dip-switches 1 and 6 of the SW1 row. 6) Power up the coin acceptor: after few seconds, triple coil activation will confirm that the coins/channels that have been manually programmed have been cancelled, and that the previous configuration as been restored. 7) Power off the coin acceptor and set the dip-switches according to the desired operation. |

PROGRAMMING BY PC SOFTWARE

THIS IS DONE THROUGH THE K-P1C-000009 PROGRAMMING KIT AND THE "Programming sw 66 v. 2 - Alberici Coin Selector" SOFTWARE.

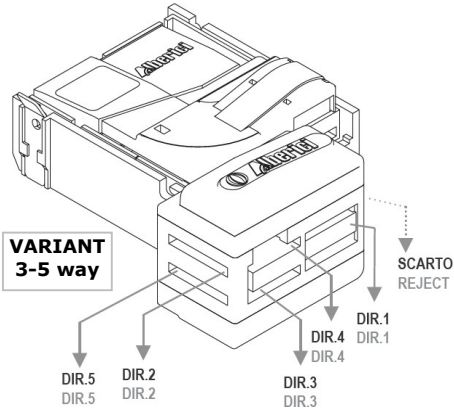
THE SOFTWARE AND ITS MANUAL, CONTAINING INSTRUCTIONS FOR KIT CONNECTIONS AND PROGRAMMING PROCEDURES, ARE BOTH AVAILABLE FREE FOR DOWNLOAD ON OUR WEB SITE

- 1 Download the application "Programming sw 66 v. 2 - AlbericiCoinSelector.exe" and the drivers for the programming interface tool from our web site <http://www.alberici.it>, "Downloads" section at the bottom of the coin acceptor page).
- 2 Install the drivers into your PC, then install the application on your desktop or in a different convenient place 
- 3 Make use of the programming-testing-power tool K-P1C-000009.
- 4 Make sure that voltage supply is off.
- 5 Connect the USB port of the PC to the USB A port of the interface.
- 6 Connect the 4pin cable between the ccTalk output of the interface and the 4-pin socket (X3) of the acceptor.
- 7 Connect the 10pin flat cable between the STD output of the interface and the 10pin socket (X1) of the acceptor:
- 8 Power the selector (green pushbutton on the interface tool).
- 9 Launch the application and modify the acceptor's data appearing on the screen according to your needs. The Instruction Manual is available on the same page of our web site from where you have downloaded the software.
- 10 Download the new configuration to the coin acceptor, switch power supply to selector off and then on again.
- 11 Test that operation corresponds to the new configuration programmed.
- 12 Switch power off and disconnect the selector from the PC USB port.



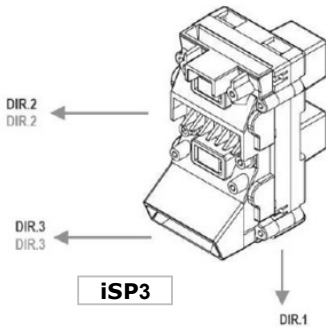
DRIVING THE NL3, NS3, VARIANT, and iS SORTERS

To attribute the proper sorting commands, please refer to the output paths shown below:



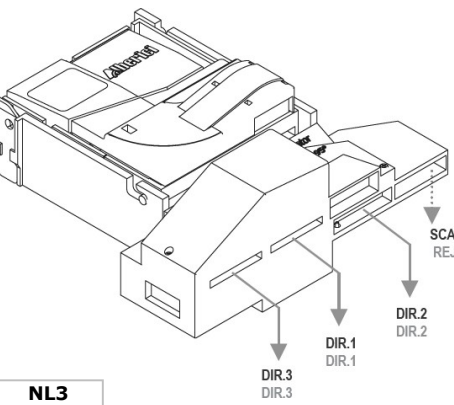
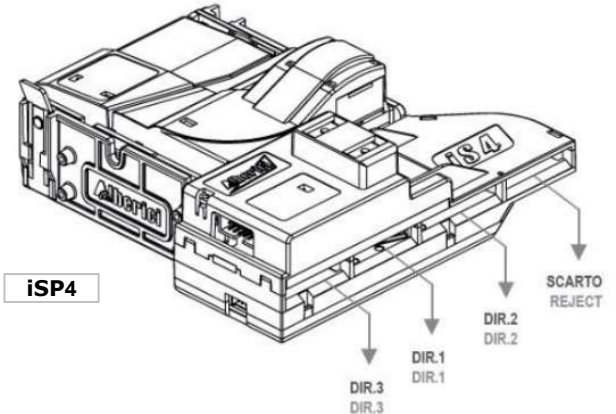
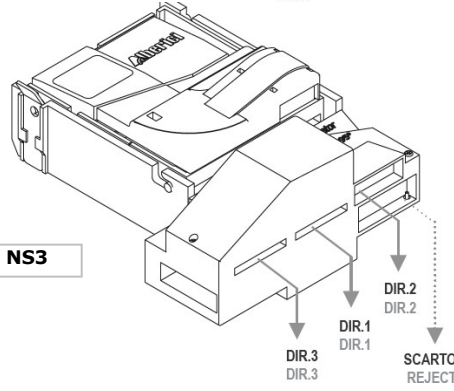
| SORTING COMMANDS | | |
|-------------------|-----------|-----------------------|
| CCTALK SORTERS | | PULSE SORTERS |
| Dir 1 = cctalk 01 | activates | Dir 1 = no OUT |
| Dir 2 = cctalk 02 | activates | Dir 2 = OUT 5 |
| Dir 3 = cctalk 03 | activates | Dir 3 = OUT 6 |
| Dir 4 = cctalk 06 | activates | Dir 4 = OUT 4 + OUT 5 |
| Dir 5 = cctalk 07 | activates | Dir 5 = OUT 4 + OUT 6 |

SOCKETS AVAILABLE ON THE SORTERS



| 10 PIN CCTALK | | | |
|---------------|---|----|-------|
| DATA | 1 | 2 | NC |
| NC | 3 | 4 | GND |
| NC | 5 | 6 | NC |
| +24 V | 7 | 8 | GND |
| NC | 9 | 10 | +24 V |

| 4 PIN CCTALK (X3) | | |
|-------------------|---|--|
| DATA | 1 | |
| GND | 2 | |
| NC | 3 | |
| +24 V | 4 | |



| 8 PIN SPI (X2) | | | |
|----------------|---|---|------------|
| DATA OUT | 1 | 2 | CLOCK |
| SELECT | 3 | 4 | DATA IN |
| GND LOGIC | 5 | 6 | N.C. |
| GND | 7 | 8 | +12/24 Vdc |

Please Note: SPI mode is available only on sorters iS1 and iS4, and it is supported only by the AL66 FG coin acceptors (see Section 7 for SPI commands)

6. CCTALK Communication Protocol

AL66 MODIFIABLE CCT HEADERS

ccTalk® communication protocol is the Money Controls serial communication protocol for low speed control networks. It was designed to allow the interconnection of various cash handling devices (*Hoppers, Bill validators, Coin selectors etc.*), mostly in AWP and gaming Industry, but also in other devices that use those components.

cctalk® is an open standard. All documentation is available at web site: www.cctalk.org.

6. Serial cctalk communication (English language)

New generation of coin selectors AL55 or AL66 use **cctalk®** communication protocol. This protocol was developed by company Emulator M (ex. Coin Controls) to enable connection of different peripheral devices (Coin selectors, Hoppers(*pay out device*), Banknote readers etc.) in small network.

Protocol is mostly used in gaming and casino machines but it can be implemented in any other types of machines that use same type of devices. It is public protocol and free to use. Pls. find documentation on: www.cctalk.org.

Communication protocol of ALBERICI coin selectors AL55/66 complies with generic specification 4.4

6.1 Communication specifications

Serial communication was derived from RS232 standard.

It is low data rate NRZ (*Non Return to Zero*) asynchronous communication with:

Baud rate 9600, 1 start bit, 8 data bits, no parity, 1 stop bit.

RS232 handshaking signals (*RTS, CTS, DTR, DCD, DSR*) are not supported.

Message integrity is controlled by means of checksum calculation.

6.1.1 Baud rate

The baud rate of 9600 was chosen as compromise between cost and speed.

Timing tolerances is same as in RS232 protocol and it should be less than 4%.

6.1.2 Voltage level

To reduce the costs of connections the "Level shifted" version of RS232 is used.

The idle state on serial connector is 5V, and active state is 0V.

Mark state (*idle*) +5V nominal from 3.5V to 5V

Space state (*active*) 0V nominal from 0.0V to 1.0V

Data I/O line is "open collector" type, so it is possible to use device in systems with different voltages.

6.1.3 Connection

The connection of Coin selector to network is achieved by means of 4 pole JST connector (*standard type 7*). Connector is used for power supply and communication as well. For schematics and connector appearance see image1.



Image 6.1 communication connector

Recommended peripheral connector is: **JST B 4B-XH-A with crimping contacts SXH-001T-P0.6**

6.2 Message structure

Each communication sequence consists of two message string.

Message string in case of simple checksum use is structured as follows:

[Destination address]

[Nr. of data bytes]

[Source address]

[Header]

[Data 1]

...

[Data n]

[Checksum]

There is an exception of message structure when device responds to instruction Address poll and Address clash5. The responds consists of only one byte representing address delayed for time proportional to address value. For CRC checksum case format is:

[Destination address]

[Nr. of data bytes]

[CRC 16 LSB]

[Header]

[Data 1]

...

[Data n]

[CRC 16 MSB]

6.2.1 Address

Address range is from address 0 to address 255. Address 0 is special case or so called "broadcast" address and address 1 is default host address.

Table 6.1 shows the recommended address values of different devices.

| Device category | Address | Additional addr. | Note |
|-----------------|---------|------------------|--|
| Coin Acceptor | 2 | 11 - 17 | Coin validator, coin selector, coinmech... |
| Payout | 3 | 4 - 10 | Hopper |
| Bill validator | 40 | 41 - 47 | Banknote reader |
| Card Reader | 50 | | |
| Display | 60 | | Alphanumeric LC display |
| Keypad | 70 | | |
| Dongle | 80 | 85 | Safety equipment |
| Meter | 90 | | Replacement for el.mec. counters |
| Power | 100 | | Power supply |
| Printer | 110 | | Ticket printing |
| RNG | 120 | | Random Number Generator |
| | | | |

Table 6.1 Standard address for different types of devices

Address for Alberici Hopper is factory set as 3; the user can change the default address by using the MDCES commands Address change or Address random or by setting Hopper dip-switches. For details see cctalk42-2.pdf, Address poll.

6.2.2 Number of data byte

Number of data byte in each transfer could be from 0 to 252.

Value 0 means that there are no data bytes in the message, and total length of message packet will be 5 bytes.

Although theoretically it will be possible to send 255 bytes of data because of some limitations in small micro controllers the number is limited to 252 (252 bytes of data, source address, header and checksum: total of 255 bytes).

6.2.3 Command headers (Instructions)

Total amount of cctalk command header is 255 with possibility to add sub-headers using headers 100, 101, 102, 103.

Header 0 stands for **ACK** (*acknowledge*) replay of device to host.

Header 5 stands for **NAK** (*No acknowledge*) replay of device to host.

Header 6 is **BUSY** replay of device to host.

In all three cases no data bytes are transferred. Use of ACK and NAK headers are explained later on, for each specific message transfer.

Commands are divided in to several groups according to application specifics:

- Basic general commands
- Additional general commands
- Commands for Coin acceptors
- Commands for Bill validators
- Commands for Payout mechs
- MDCES commands

ALBERICI Coin selectors AL55166 use total of 557 instructions-headers.

6.2.4 Data

There is no limitation in use of data formats. Data could be BCD (*Binary Coded Decimal*) numbers, Hexa numbers or ASCII strings. Interpretation as well as format is specific to each header use, and will be explained in separate chapter.

6.4.5 Checksum

Message integrity during transfer is checked by use of simple zero checksum calculation.

Simple checksum is made by 8 bit addition (modulus 256) of all the bytes in the message. If message is received and the addition of all bytes are non-zero then an error has occurred (See Error handling).

For noisy environment or higher security application it is possible to use more complex, 16 bit CRC CCITT checksum based on a polynomial of:

$x^{16} + x^{12} + x^5 + 1$ and initial value of CRC register **0x0000**.

Hopper are using simple checksum, but they could be set to operate with CRC-16 checksum on customer demand.

6.5 Timing specification

The timing requirements of cctalk are not very critical but there are some recommendations (i.e. more than 100 mili sec for solenoid testing).

6.5.1 Time between two bytes

When receiving bytes within a message packet, the communication software must wait up to **50 ms** for next byte if it is expected. If time out occurs, the software should reset all communication variables and get ready to receive next message. The inter-byte delay during transmission should be ideally **less than 2 ms** and **not greater than 10 ms**.

6.5.2 Time between command and replay

The time between command and reply is dependent on application specific for each command. Some commands return data immediately, and maximum time delay should be within **10 ms**. Other commands that must activate some actions in device may return reply after the action is finished

6.5.3 Start-up time

After the power-up sequence Hopper should be ready to accept and answer to a cctalk message within time period of less than 250 ms. During that period all internal check-up and system settings must be done, and Hopper should be able works fine.

6.4 Error handling

If slave device receive the message with bad checksum or missing data no further action is taken and receive buffer will be cleared. Host software should decide to re-transmit message immediately or after a fixed amount of time. In case when host receive message with error it has same options.

6.5 Command headers

Command header set, that host could use in communication with coin selectors AL55 and AL66 is given in the table 6.2.

| | | | | | |
|-----|----|--|-----|----|--------------------------------|
| 254 | FE | Simple poll | 228 | E4 | Modify master inhibit status |
| 253 | FD | Address poll (broadcast) | 227 | E3 | Request master inhibit status |
| 252 | FC | Address clash (no broadcast) | 226 | E2 | Request insertion counter |
| 252 | FC | Address clash (broadcast) | 225 | E1 | Request acceptance counter |
| 251 | FB | Address change (not supported in coin acceptors) | 210 | D2 | Modify sorter paths |
| 249 | F9 | Request polling priority | 209 | D1 | Request sorter paths |
| 248 | F8 | Request status | 197 | C5 | Calculate ROM checksum |
| 246 | F6 | Request manufacturer id | 196 | C4 | Request creation date |
| 245 | F5 | Request equipment category id | 195 | C3 | Request last modification date |
| 244 | F4 | Request product code | 194 | C2 | Request reject counter |
| 242 | F2 | Request serial number | 193 | C1 | Request fraud counter |
| 241 | F1 | Request software revision | 192 | C0 | Request build code |
| 240 | F0 | Test solenoids | 188 | BC | Request default sorter path |
| 237 | ED | Read input lines | 184 | B8 | Request coin id |
| 236 | EC | Read opto states | 170 | AA | Request base year |
| 232 | E8 | Perform self test | 4 | 04 | Request comms revision |
| 231 | E7 | Modify inhibit status | 3 | 03 | Clear comms status variables |
| 230 | E6 | Request inhibit status | 2 | 02 | Request comms status variables |
| 229 | E5 | Read buffered credit or error codes | 1 | 01 | Reset device |

Table 6.2 cctalk instruction header list

Command headers are divided in to 3 different groups:

- Common command headers
- Coin acceptor command headers
- MDCES command headers

6.5.1 Common command headers

Common commands are used in all type of devices to detect their presence on cctalk network or to describe them. Information like: manufacturer or product type id, serial number, different settings etc. are transmitted to host.

6.5.1.1 Command 254 [hexFE], Simple poll

The fastest way for host to detect all attached devices in cctalk network.

Addressed device-coin selector responds with ACK (*Acknowledge*).

If within predicted amount of time Coin selector does not responds coin selector is probably not connected, powered or simple not working properly.

Message format is:

Host sends: [Dir] [00] [01] [FE] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk]

As coin selector default address is 2, example of message string is:

Host sends: [02] [00] [01] [FE] [FF]

Coin s. responds: [01] [00] [02] [00] [FD] ACK message

6.5.1.2 Command 246 [hexF6], Request manufacturer ID

Coin selector responds with ASCLL string representing manufacturer name.

Message format is:

Host sends: [Dir] [00] [01] [F6] [Chk]

Coin s. responds: [01] [Nr.b] [Dir] [00] [a1] [a2] [an] [Chk]

Nr. b is number of data bytes-characters sent by coin selector, and a1 to an are ASCII characters. For **Alberici** coin selector example of message string is:

Host sends: [02] [00] [01] [F6] [07]

Coin s. responds: [01] [08] [02] [00] [41][6C][62][65][72][69][63][69] [DA]

6.5.1.3 Command 245 [hexF5], Request equipment category ID

Responds to command header is standardized name for coin selectors, coin validators or coin mechs. Coin selector responds with ASCII string of characters representing standardized name for that type of device **Coin Acceptor**.

Message format is:

Host sends: [Dir] [00] [01] [F5] [Chk]

Coin s. responds: [01] [0D] [Dir] [00] [43][6F][69][6E][20][41][63][63][65][70][74][6F][72] [Chk]

Number of data byte is always 13, hex [0D].

Example of message string for coin selector(*address 2*) is:

Host sends: [02] [00] [01] [F5] [08]

Coin s. responds: [01] [0D] [02] [00] [43][6F][69][6E][20][41][63][63][65][70][74][6F][72] [16]

6.5.1.4 Command 244 [hexF4], Request product code

Coin selector responds with ASCII string of character, representing the factory type of coin selector. For ALBERICI coin selectors of new generation possible response will be:

- AL55V1, AL55K1, AL55I1

- AL66V2, AL66K3, AL66I3

In special version for italian gambling machines response is allways **AL06V-c** .

Host sends: [Dir] [00] [01] [F4] [Chk]

Coin s. responds: [01] [07] [Dir] [00] [a1][a2] [a7] [Chk]

Number of data bytes sent by coin selector is 6 or 7, hex [07].

Example of message string for coin selector(*address 2*) type **AL06V-c** is:

Host sends: [02] [00] [01] [F4] [09]

Coin s. responds: [01] [07] [02] [00] [41][4C][30][36][56][2D][63] [1D]

6.5.1.5 Command 242 [hexF2], Request serial number

Coin selector responds with three byte serial number. Message format is:

Host sends: [Dir] [00] [01] [F2] [Chk]

Coin s. responds: [01] [03] [Dir] [00] [Serial 1 - LSB] [Serial 2] [Serial 3 - MSB] [Chk]

Serial 1 – first data byte sent is LSB of serial number.

Example of message string for coin selector(address 2) with serial number: 1234567

(hex [BC][61][4E]) is:

Host sends: [02] [00] [01] [F2] [0B]

Coin s. responds: [01] [03] [02] [00] [4E][61][BC] [8F]

6.5.1.6 Command 241 [hexF1], Request software revision

Coin selector return ASCII string of character representing software version and revision.

Message format is:

Host sends: [Dir] [00] [01] [F1] [Chk]

Coin s. responds: [01] [Nr.b] [Dir] [00] [a1] [a2]. ... [an] [Chk]

Number of data bytes in ASCII string is not limited and each producer has it's own system of labelling. Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [F1] [0C]

Coin s. responds: [01] [09] [02] [00] [75][31][2E][30][20][70][31][2E][30][2E][30] [71]

Coin selector response is 'u1.0 p1.0.0'.

ALBERICI coin selectors has program firmware label divided in two parts.

First label **u** is for protected FLASH memory program(*monitor program*) revision.

First digit is for major changes and second for minor changes. In this case it is **u1.0**.

Second label is revision of main program FLASH memory.

Main program software revision labelling use 3 digits. First most significant digit is for major software changes, second is for minor software changes and third for "bug" correction. In this case it is **u1.0.0**.

6.5.1.7 Command 197 [hexC5], Calculate ROM checksum

Coin selector responds with four bytes of micro controller internal memory checksum. First two bytes are program ROM CRC and the second is data EEPROM CRC. Any changes in program or data will change the responds of coin selector.

Message format is:

Host sends: [Dir] [00] [01] [C5] [Chk]

Coin s. responds: [01] [4] [Dir] [00] [CRC1-H][CRC1-L] [CRC2-H] [CRC2-L] [Chk]

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [C5] [38]

Coin s. responds: [01] [04] [02] [00] [D9][2A][7E][79] [96]

6.5.1.8 Command 192 [hexC0], Request build code

Coin selector responds with ASCII string of character representing it's hardware version and revision. Last revision of printed circuit board for coin selectors AL55/66 is: **AL66 V1.0**.

Message format is:

Host sends: [Dir] [00] [01] [C0] [Chk]

Coin s. responds: [01] [Nr.b] [Dir] [00] [a1] [a2]. ... [an] [Chk]

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [C0] [3D]

Coin s. responds: [01] [09] [02] [00] [41][4C][2D][30][35][20][56][35][30] [FA]

6.5.1.10 Command 4 [hex04], Request comms revision

Coin selector responds with three byte data information about level of cctalk protocol implementation, major and minor revision. Message format is:

Host sends: [Dir] [00] [01] [04] [Chk]

Coin s. responds: [01] [03] [Dir] [00] [Level] [Mag.rev.] [min. rev.] [Chk]

Example of message string for coin selector(address 2) with level of implementation 1, cctalk protocol issue 4.4 is:

Host sends: [02] [00] [01] [04] [F9]

Coin s. responds: [01] [03] [02] [00] [01][04][04] [F1]

6.5.1.11 Command 3 [hex03], Clear comms status variables

After acceptance of command header 3, coin selector clears all three bytes of communication errors counters and responds with ACK message. Message format is:

Host sends: [Dir] [00] [01] [03] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK message

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [03] [FA]

Coin s. responds: [01] [00] [02] [00] [FD] ACK message

10 Details of description see in public document cctalk44-2.pdf

11 Address change, Address random

6.5.1.12 Command 2 [hex02], Request comms status variables

Coin selector responds with three byte data representing communication errors.

First byte is receive time out counter, second byte is number of ignored receive bytes (Number of receive buffer overflow bytes), and third byte is number of checksum errors. Message format is:

Host sends: [Dir] [00] [01] [02] [Chk]

Coin s. responds: [01] [03] [Dir] [RxErr1] [RxErr2] [RxErr3] [Chk]

Example of message string for coin selector(address 2) with no errors is:

Host sends: [02] [00] [01] [02] [FB]

Coin s. responds: [01] [03] [02] [00] [00] [00] [FA]

6.5.1.13 Command 1 [hex01], Reset device

After acceptance of Reset command, coin selector execute software reset and clear all variables in RAM or set them at default value, including different counters and credit buffer. ACK message is sent before reset of coin selector. Host software must set again:

- inhibit state
- sorter path
- master inhibit (*if necessary*)

Message format is:

Host sends: [Dir] [00] [01] [01] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK message

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [01] [FC]

Coin s. responds: [01] [00] [02] [00] [FD] ACK message

Host software must wait at least **100 ms**, to continue communication with coin selector after reset instruction!

6.5.2 Coin acceptor specific command headers

Coin selectors use some specific commands, mostly for control of coin input, acceptance and direction (Sorter control commands). Some commands are shared with other device like banknote reader or payout device.

6.5.2.1 Command 249 [hexF9], Request polling priority

Basic principle of detecting credit input or eventual errors from coin selector is sequential polling1 (Reading memory buffer from coin selector). Coin selectors due to differences in mechanical and electrical construction has different acceptance speed. All events are registered in memory buffer with limited size1 (Five stage double byte memory buffer). To avoid credit loss, host must read coin selector credit buffer within limited time period.

Coin selector has internal mechanism to block the coin acceptance and registration of all events if polling time elapse. For ALBERIC1 coin selector acceptance speed is from 3 to 4 coins per second (Dependant on mechanical type of coin selector - S type is faster than V type - and coin.

Considering that it is possible to register 5 event in the buffer, the adequate polling time will be about 1 sec. Because of necessity to register even "close" and non accepted coins polling time must be even shorter.

For ALBERIC1 coin selectors AL55/66 using cctalk interface, poll time is set to 500 ms.

Coin selectors that use standard 10 pole interface are not necessary to poll. In that case polling time unit is set to 0(*no polling*)! Minimum time for polling must not be shorter than overall message time. It is 38 ms for coin selector with response time 2ms and byte gap 1 ms.

Coin selector responds to command with two bytes of data. First byte is poll time unit and second is polling time value (For details see, cctalk44-2.pdf). Message format is:

Host sends: [Dir] [00] [01] [F9] [Chk]

Coin s. responds: [01] [01] [Dir] [Time] [Chk]

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [F9] [04]

Coin s. responds: [01] [02] [02] [00] [02] [32] [C7]

First byte **02** is unit **x10ms** , and second byte is time value **hex32 = 50**.

Polling time is calculated as:

T = 10 x 50 = 500 ms

6.5.2.2 Command 248 [hexF8], Request Status

ALBERICI coin selectors has no additional COS (Coin On String) and return mechanism.

Response to that command is always hex[00], coin selector Ok.

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [F8] [05]

Coin s. responds: [01] [01] [02] [00] [00] [FC]

6.5.2.3 Command 243 [hexF3], Request data base version

The responds to that command is version of coin data base. Version of data base is important for coin selectors with remote programming support. For all ALBERICI coin selectors type AL55/66 current data base version is 00.

Message format is:

Host sends: [Dir] [00] [01] [F3] [Chk]

Coin s. responds: [01] [01] [Dir] [Ver.] [Chk]

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [F3] [0A]

Coin s. responds: [01] [01] [02] [00] [00] [FC]

6.5.2.4 Command 240 [hexF0], Test solenoids

Host sends one byte mask to determinate which solenoid must be tested.

Coin selector accept gate solenoid or sorter solenoid will be switched on for period of 100ms and after that, ACK message will be transmitted. Message format is:

Host sends: [Dir] [01] [01] [F0] [Mask.] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK

Example of message string for coin selector(address 2) acceptance gate test is:

Host sends: [02] [01] [01] [F0] [01] [0B]

Coin s. responds: [01] [00] [02] [00] [FD] Single click -> 100 ms, ACK

Bit position for output that is used to drive sorter coil are:

bit 0 = accept gate coil

bit 1 = sorter coil "A"(out 6/pin 4)

bit 2 = sorter coil "B"(out 5/pin 3)

bit 3 = sorter coil "C"(out 4/pin 10)

If output selected with bit in mask is not programmed for sorter activation it will not be activated but coin selector will still response with ACK.

6.5.2.5 Command 238 [hexEE], Test output lines

Host sends one byte mask to determinate which output line must be tested.

Coin selector output line that corresponds to bit set in the mask will be pulsed for 100 ms and after that, ACK message will be transmitted. Message format is:

Host sends: [Dir] [01] [01] [EE] [Mask.] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK

Example of message string for coin selector(address 2) first output(pin 7) is:

Host sends: [02] [01] [01] [EE] [01] [0D]

Coin s. responds: [01] [00] [02] [00] [FD] Single pulse out 1 -> 100 ms, ACK

Bit positions for output test are:

- bit 0 Output 1(*pin 7*)
- bit 1 Output 2(*pin 8*)
- bit 2 Output 3(*pin 9*)
- bit 3 Output 4(*pin 10*)
- bit 4 Output 5(*pin 3*)
- bit 5 Output 6(*pin 4*)
- bit 6 Output 7(*pin 5*)
- bit 7 Not used

Unused output (*not programmed*) will not be turned on, but message ACK will be returned.

6.5.2.6 Command 237 [hexED], Read input lines

Coin selector responds with two data byte representing the state of DIP-switches and state of inputs In1(*pin 6*) and In2(*pin 5*), if In2 is programmed as input. ALBERICI coin selectors has one or two banks of DIP-switches for various data or operating modes setting. First data byte is state of first DIP-switch(*bank 1*) and In1, while second represent second DIP-switch(*bank 2*) and In2. LSB is first switch in bank and MSB is state of input. Switch closed state is represented with logic "1", and input active state is logic "1". Message format is:

Host sends: [Dir] [00] [01] [ED] [Chk]

Coin s. responds: [01] [02] [Dir] [Mask1] [Mask2] [Chk]

Example of message string for coin selector(*address 2*), with all switches "off" and inputs not active is:

Host sends: [02] [00] [01] [ED] [10]

Coin s. responds: [01] [02] [02] [00] [00] [00] [FB]

Example of message string for coin selector(*address 2*), with all switches "on" and input 1(*inhibit acceptance*) active is:

Host sends: [02] [00] [01] [ED] [10]

Coin s. responds: [01] [02] [02] [00] [BF] [00] [3C]

6.5.2.7 Command 236 [hexEC], Read opto states

Coin selector responds with one data byte representing the state of opto pairs.

ALBERICI coin selectors has up to 3 pairs of optical sensor (notice: in some case group could contain more than one opto pairs) for detection of coin position, speed and direction and 2 pairs of opto sensors for diameter measurement.

Bit position for opto pairs are:

- bit 0 Diam. measure opto 1
- bit 1 Diam. measure opto 2
- bit 2 Control opto 1
- bit 3 Control opto 2
- bit 4 Control opto 3
- bit 5 Not used
- bit 6 Not used
- bit 7 Not used

Control opto sensor 2 is called "credit" opto sensor exist in all version of coin selectors and it is placed after the acceptance gate. Other pairs are optional and some coin selectors has 2 and some 3 control optical pairs. Number of control pairs make part of coin selector type label. For example coin selector type AL66V2 has 2 control opto sensor pairs. The unused bits or non existing optical sensors are always read as 0.

Interruption of light barrier of opto sensor corresponds to bit value 1.

Message format is:

Host sends: [Dir] [00] [01] [EC] [Chk]

Coin s. responds: [01] [01] [Dir] [Mask.] [Chk]

Example of message string for coin selector(*address 2*) with opto sensors cleared is: Host sends: [02] [00] [01] [EC] [11]

Coin s. responds: [01] [01] [02] [00] [00] [FC]

6.5.2.8 Command 233 [hexE9], Latch output lines

This instruction is similar to instruction 238. Host sends one byte mask to determinate which output line must be activated(*latch*). ACK message will be transmitted immediate.

Coin selector output line that corresponds to bit set in the mask will be latched and active till reset or new instruction with bit cleared is sent. Message format is:

Host sends: [Dir] [01] [01] [E9] [Mask.] [Chk]
 Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK
 Example of message string for coin selector(address 2) first output(pin7) is:
 Host sends: [02] [01] [01] [E9] [01] [12]
 Coin s. responds: [01] [00] [02] [00] [FD] Latch out 1 -> ACK

6.5.2.9 Command 232 [hexE8], Perform self-test

IMPORTANT WARNING: send only after at least 2 seconds have elapsed from power-on or from reset command.

Coin selector responds to command with one or two bytes of data according to table 6.3. First byte is fault code and second is optional data, usually representing fault sensor number(from 1 to 3).

| Code | Fault | Optional data | Comment |
|------|----------------------------|---------------|-----------------------|
| 0 | OK No fault detected | - | - |
| 2 | Fault on inductive sensor | Sensor number | - |
| 3 | Fault on credit sensor | - | Control opto sensor 2 |
| 6 | Fault on diameter sensor | - | - |
| 18 | Fault on reject sensor | - | Control opto sensor 3 |
| 33 | Power supply out of limits | - | - |
| 34 | Temperature out of limit | - | Optional |
| 255 | Unspecified fault code | - | - |

Table 6.3 Fault codes for AL55/66 coin selectors

Inductive sensor numbers are:

01 Upper inductive sensor 02 First lower inductive sensor 03 Second lower inductive sensor

Message format is:

Host sends: [Dir] [00] [01] [E8] [Chk]
 Coin s. responds: [01] [01|02] [Dir] [Fault c.][Data opt.] [Chk]
 Example of message string for coin selector(address 2) with no fault detected is:
 Host sends: [02] [00] [01] [E8] [15]
 Coin s. responds: [01] [01] [02] [00] [00] [FC] No fault detected
 Ex. of message string for coin selector (addr. 2) with first lower sensor fault detected is:
 Host sends: [02] [00] [01] [E8] [15]
 Coin s. responds: [01] [02] [02] [00] [02][02] [F7] Fault on first lower sensor detected

6.5.2.10 Command 231 [hexE7], Modify inhibit status

With this command host is able to inhibit the acceptance of some or all coins.

Acceptance or inhibition is set with a two byte mask sent by host.

Bits from 0 do 15 determinate coin positions from 1 to 16. Positions are sent by coin selector during reading credit buffer or error codes (header 229).

Number of coin channels in new ALBERICI coin selectors AL55/66 is same as number of position(16).

Message format is:

Host sends: [Dir] [02] [01] [E7] [LSB Mask.] [MSB Mask.] [Chk]
 Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK
 Example of message string to enable all position for coin selector(address 2) is:
 Host sends: [02] [02] [01] [E7] [FF] [FF] [16]
 Coin s. responds: [01] [00] [02] [00] [FD] ACK

After that all programmed coins will be enabled. Command has no effect on coin position that are not programmed.

Initially coin channels could be programmed with acceptance enabled or disabled.

For coin selectors that are using only cctalk interface, all coins position must be initially inhibited!

6.5.2.11 Command 230 [hexE6], Request inhibit status

Coin selector responds with two byte data that corresponds to inhibit state mask for all 16 positions of coin. If bit value is 1 acceptance of coin in that position is enabled. If bit value is 0 coin is inhibited. Message format is:

Host sends: [Dir] [02] [00] [E6] [Chk]
 Coin s. responds: [01] [02] [Dir] [00] [LSB Mask.] [MSB Mask.] [Chk]
 Example of message string for coin selector(address 2) AL06V-c (Coin selector for Italian gambling machines) after power-up or reset is:

Host sends: [02] [00] [01] [E6] [17]

Coin s. responds: [01] [02] [02] [00] [00] [00] [FB]

Example of message string for coin selector(address 2) with programmed positions from 1 to 6, after receiving command to enable acceptance of all 16 position is:

Host sends: [02] [00] [01] [E6] [17]

Coin s. responds: [01] [02] [02] [00] [3F] [00] [BC]

First byte represent the mask for coin positions 1 to 8 and second for 9 to 16.

Coin channels(positions) that are not programmed are always represented as zero bit!

6.5.2.12 Command 229 [hexE5], Read buffered credit or error codes

This is the most important command used by host to detect import of coins into a machine and to report eventual errors. As previously mentioned coin selectors store all events in volatile memory called credit buffer. Buffer has 5 level and use two bytes for each event. In first byte coin position or coin value is stored, if coin selector uses CVF (Coin Value Format). The second byte point to a sorter path or indicate error code.

If during coin acceptance any error occurs, stored value of coin position is 0, hex [00].

Error codes supported in ALBERICI coin selectors AL55/66 are shown in table 6.4.

| Code d. | Code h. | Error | Coin rejected |
|---------|---------|--|---------------|
| 0 | 00 | Null event | No |
| 1 | 01 | Reject coin (not recognized) | Yes |
| 2 | 02 | Inhibited coin (master inhibit) | Yes |
| 3 | 03 | Multiple window (fraud or similar coin) | Yes |
| 5 | 05 | Validation (measuring) time out | Yes |
| 6 | 06 | Credit sensor (recognition to opto 2) time out | Possible |
| 8 | 8 | Second close coin | Yes/both |
| 16 | 10 | Credit sequence error (Yo-yo) | No |
| 18 | 12 | Coin to fast (opto 2 minimum time not elapsed) | No |
| 19 | 13 | Coin to slow (opto 2 time out) | No |
| 128 | 80 | Inhibited coin (position 1) | Yes |
| ... | ... | Inhibited coin (position n) | Yes |
| 143 | 8F | Inhibited coin (position 16) | Yes |
| 255 | FF | Unspecified alarm code | - |

Table 6.4 Acceptance error codes

Coin selectors also use one eight bit counter (Event counter) that is incremented each time a new coin is detected. At the same time data two positions are shifted to the right in coin credit buffer. When counter reaches the value of 255 it toggle to a value 1 and continue to increment on each event. Event counter is set to value "0" after each power-up or acceptance of reset command. The first two byte (LSB) in coin credit buffer always contain the data of last event. Host software must read event counter and coin credit buffer data in period short enough to prevent the loss of coin data (See command 249 Request polling priority). Message format is:

Host sends: [Dir] [00] [00] [E5] [Chk]

Coin s. responds: [01][0B] [Dir] [00] [Ev.cnt.][coin code 1][dir/err] [coin code 2][dir/err] . . .
 . . . [coin code 5][dir/err] [Chk]

Examples of message string for coin selector(address 2) after coin insertions:

Host sends: [02] [00] [00] [E5] [18] Polling minimum each 500 ms

Coin s. responds: [01] [0B] [02] [00] [00][00][00][00][00][00][00][00][00][00] [F2]

The responds after power-up or reset

Coin s. responds: [01] [0B] [02] [00] [01][01][02][00][00][00][00][00][00][00][00] [EE]

First event, coin position 1, sorter path 2 accepted

Coin s. responds: [01] [0B] [02] [00] [02][02][01][01][02][00][00][00][00][00][00] [EA]

Second event, coin position 2, sorter path 1 accepted

Coin s. responds: [01] [0B] [02] [00] [03][00][02][02][01][01][02][00][00][00][00] [E7]

Third event, coin rejected due to master inhibit active

Coin s. responds: [01] [0B] [02] [00] [04][00][83][00][02][02][01][01][02][00][00] [63]

Forth event, coin position 4 inhibited and rejected

From example we can notice shifting of data in the coin credit and error buffer and increment of event counter.

6.5.2.13 Command 228 [hexE4], Modify master inhibit status

This command is used to inhibit acceptance of all coins and has same effect as command modify inhibit status with sent with two bytes of zeros. Host sends only one byte of data. If first bit (*LSb*) is set to "0" coin selector is inhibited. Bits 1 to 7 has no influence to coin selector. Message format is:

Host sends: [Dir] [01] [01] [E4] [Mask.] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK

Initially coin selectors are programmed with acceptance enabled. Change is stored in RAM location.

On customer demand it is possible to set inhibition as default .

Example of message string to inhibit the acceptance for coin selector(*address 2*) is:

Host sends: [02] [01] [01] [E4] [00] [18]

Coin s. responds: [01] [00] [02] [00] [FD] ACK

After that coin selector acceptance will be inhibited till reset or next instruction that will change master inhibit status.

6.5.2.14 Command 227 [hexE3], Request master inhibit status

Coin selector responds with one byte data information of main inhibit status.

Only first (*LSb*) bit is used. If bit 0 is "1" acceptance is enabled, and if bit 0 is "0" coin selector is inhibited and acceptance is disabled. Other bits has no meaning and always read as "0". Message format is:

Host sends: [Dir] [00] [00] [E3] [Chk]

Coin s. responds: [01] [01] [Dir] [00] [Mask.] [Chk]

Example of message string for coin selector(*address 2*) after power-up is:

Host sends: [02] [00] [01] [E3] [1A]

Coin s. responds: [01] [01] [02] [00] [01] [FB] Acceptance enabled (*default*)

Example of message string for coin selector(*address 2*) after activation of master inhibit27 is:

Host sends: [02] [00] [01] [E3] [1A]

Coin s. responds: [01] [01] [02] [00] [00] [FC] Coin selector inhibited

6.5.2.15 Command 226 [hexE2], Request insertion counter

Coin selector responds with three bytes of insertion counter data.

First byte is LS byte of three byte counter in RAM. Insertion counter is set to zero after power up or reset command. It is incremented each time a new coin is inserted in to coin acceptor. Message format is:

Host sends: [Dir] [00] [00] [E2] [Chk]

Coin s. responds: [01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]

Example of message string for coin selector(*address 2*) after power-up is:

Host sends: [02] [00] [01] [E2] [1B]

Coin s. responds: [01] [03] [02] [00] [00] [00] [FA]

6.5.2.22 Command 210 [hexD2], Modify sorter paths

With this command host is able to change direction of coins in sorter if sorter is supported. Host sends two bytes of data to select the coin position and sorter path (*direction of exit*). First byte of data (*LSB*) represent coin position and second byte of data point to sorter path. ALBERICI coin selectors has support for most existing sorters that has direct drive of coils from coin selector with open collector transistor. Most common are 3 or 4 way sorter with two coils -(Maximum current consumption for each coil is 500 mA, but recently 5 way sorters with 3 coils are in use3 (ex. 5-way VARIANT sorter from ALBERICI).

Message format is:

Host sends: [Dir] [02] [01] [D2] [Coin pos.] [Sort.Path] [Chk]

Coin s. responds: [01] [00] [Dir] [00] [Chk] ACK if sorter path is possible to set

Coin s. responds: [01] [00] [Dir] [05] [Chk] NAK if coin selector does not support setting

Initially all coin position has sorter paths set to direction 1 hex[01]. If sorter is not supported, sorter path is set initially to 0 hex[00]!

If host sends command to modify sorter path that is not existent or for coin not programmed, the coin selector will respond with message NAK.

Example of message string for coin selector(*address 2*) redirection of coin **pos. 1** in to **path 2** is:

Host sends: [02] [02] [01] [D2] [01] [02] [26]

Coin s. responds: [01] [00] [02] [00] [FD] ACK

After acceptance of command, accepted coins with position 1 will exit in direction 2 of the sorter. The path or direction 1 is usually one without activation of any coil.

It is possible to program a different coil activation schematics by setting the sorter type.

6.5.2.23 Command 209 [hexD1], Request sorter paths

Host send one byte of coin position and coin selector responds with one byte of sorter path. Message format is:

Host sends: [Dir] [01] [00] [D1] [Coin pos.] [Chk]

Coin s. responds: [01] [01] [Dir] [00] [Sort.Path] [Chk]

Example of message string for coin selector(*address 2*) for initial sorter path 1 of coin position 1:

Host sends: [02] [01] [01] [D1] [01] [2A]

Coin s. responds: [01] [01] [02] [00] [01] [FB]

Example of message string for coin selector (*address 2*) for sorter path 2 of coin position 1:

Host sends: [02] [01] [01] [D1] [01] [2A]

Coin s. responds: [01] [01] [02] [00] [02] [FA]

If host request sorter path for non programmed coins or non existent position (Position higher than 16), the coin selector will respond with message NAK !

6.5.2.26 Command 196 [hexC4], Request creation date

Coin selector responds with two byte of data that represent codified date of production.

Date of production is codified in so called *RTBY (Relative To Base Year)* format (For details see cctalk protocol, document cctalk44-2.pdf).

Message format is:

Host sends: [Dir] [00] [01] [C4] [Chk]

Coin s. responds: [01] [02] [Dir] [00] [LSB] [MSB] [Chk]

Example of message string for coin selector (*address 2*) with date of production 05.07.2003 is:

Host sends: [02] [00] [01] [C4] [39]

Coin s. responds: [01] [02] [02] [00] [E5] [06] [10]

ALBERICI coin selectors has date of production written in monitor part of MCU FLASH memory which is not possible to change without factory FLASH reprogramming.

6.5.2.27 Command 195 [hexC3], Request last modification date

Coin selector responds with two byte of data that represent codified date of last modification of software3 (Up-grade of FLASH program memory). Date of modification is codified also in RTBY format.

Message format is:

Host sends: [Dir] [00] [01] [C3] [Chk]

Coin s. responds: [01] [02] [Dir] [00] [LSB] [MSB] [Chk]

Example of message string for coin selector (*address 2*) with date of modification 23.07.2003 is:

Host sends: [02] [00] [01] [C3] [3A]

Coin s. responds: [01] [02] [02] [00] [F7] [06] [FE]

NOTICE: after each up-grade of coin selector program FLASH memory date will correspond to software modification date, not to the actual date of up-grade!

6.5.2.28 Command 194 [hexC2], Request reject counter

Coin selector responds with three bytes of reject counter data.

First byte is LS byte of three byte counter in RAM. Reject counter is set to zero after power up or reset command. It is incremented each time a coin is inserted but not recognized. Message format is:

Host sends: [Dir] [00] [00] [C2] [Chk]

Coin s. responds: [01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]

Example of message string for coin selector(*address 2*) after power-up is:

Host sends: [02] [00] [01] [C2] [3B]

Coin s. responds: [01] [03] [02] [00] [00] [00] [00] [FA]

6.5.2.29 Command 193 [hexC1], Request fraud counter

Coin selector responds with three bytes of fraud coins counter data.

First byte is LS byte of three byte counter in RAM. Fraud counter is set to zero after power up or reset command. It is incremented each time a coin acceptor recognize coin that is programmed as "fraud" coin (Coins with close recognition parameters sometime called "killer coin or channel"). Message format is:

Host sends: [Dir] [00] [00] [C1] [Chk]

Coin s. responds: [01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]

Example of message string for coin selector(address 2) after power-up is:

Host sends: [02] [00] [01] [C1] [3C]

Coin s. responds: [01] [03] [02] [00] [00] [00] [00] [FA]

6.5.2.30 Command 188 [hexBC], Request default sorter path

For ALBERICI coin selectors AL55/66 the default sorter path is always hex[01].

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [BC] [41]

Coin s. responds: [01] [01] [02] [00] [01] [FB]

6.5.2.32 Command 184 [hexB8], Request coin ID

Host use this command at initialization process to build table for each coin position value. If coin selector uses CVF it is obsolete command. Host sends one byte data of coin position and coin selector responds with 6 byte ASCII string of characters that describes the requested coin position.

Message format is:

Host sends: [Dir] [01] [01] [B8] [Coin pos] [Chk]

Coin s. responds: [01] [06] [Dir] [00] [a1][a2][a3][a4][a5][a6] [Chk]

Example of message string for coin selector(address 2) and coin position 1(2 Euro) is:

Host sends: [02] [01] [01] [B8] [01] [43]

Coin s. responds: [01] [06] [02] [00] [45][55][32][30][30][41] [8A] Coin 'EU200A'

For none-programmed position the ASCII string is: '.....'.

Example of message string for coin selector(address 2) and coin position 12 that is not programmed is:

Host sends: [02] [01] [01] [B8] [0C] [38]

Coin s. responds: [01] [06] [02] [00] [2E][2E][2E][2E][2E] [E3] Coin not programed

6.5.2.33 Command 176 [hexB0], Request alarm counter

Coin selector responds with one bytes of alarm counter data.

Alarm counter is set to zero after power up or reset command. It is incremented each time a coin acceptor detect any type of erroneous coin acceptance. Possible alarms: Coin direction error(Jojo), coin too slow (Coin jam) or coin too fast

Message format is:

Host sends: [Dir] [00] [00] [B0] [Chk]

Coin s. responds: [01] [01] [Dir] [00] [Cunt] [Chk]

Example of message string for coin selector(address 2) after power-up is:

Host sends: [02] [00] [01] [B0] [4D]

Coin s. responds: [01] [03] [02] [00] [00] [FC]

6.5.2.34 Command 173 [hexAD], Request thermistor reading

Some coin selectors AL66 (Coin selectors for use in extreme ambient temperature conditions, i.e. for external use) include a built-in linear temperature sensor.

Using this command is possible to read temperature on surface of coin selector PCB.

If temperature sensor is not built in coin selector will not responds to this command.

Temperature sensor is linear type, with 1 unit change for one degree Celsius change.

For 0°C value will be dec[50], for ie. 25°C it will be dec[75], for -10°C it will be

dec[40] and for 50°C it will be dec[100]. Message format is:

Host sends: [Dir] [00] [00] [AD] [Chk]

Coin s. responds: [01] [01] [Dir] [00] [Temp] [Chk]

Example of message string for coin selector(address 2) at ambient temperature of 25°C is:

Host sends: [02] [00] [01] [AD] [50]

Coin s. responds: [01] [01] [02] [00] [4B] [B1]

6.5.2.35 Command 170 [hexAA], Request base year

Coin selector responds with four byte ASCII string of character representing the base year for calculation of exact date of production. Message format is:

Host sends: [Dir] [00] [01] [AA] [Chk]

Coin s. responds: [01] [04] [Dir] [00] [a1][a2][a3][a4] [Chk]

For ALBERICI coin selectors base year is 2000.

Example of message string for coin selector(address 2) is:

Host sends: [02] [00] [01] [AA] [53]

Coin s. responds: [01] [04] [02] [00] [32][30][30][30] [37]

6.5.3 MDCES command headers

MDCES stands for **M**ulti-**D**rop **C**ommand **E**xtension **S**et, or so called Multi-drop bus commands. Multi-drop bus commands give additional functionality to systems that require change of address for devices in cctalk network. Some of commands has different message format than usual cctalk message.

Commands are: - Address poll

- Address clash

- Address change

- Address random

Because host always uses address 1 and address 0 is for broadcast message all commands that changes the address should not accept this settings.

All changes are stored in non-volatile memory, EEPROM !

6.5.3.1 Command 253 [hexFD], Address poll

This is a broadcast message used by host to determinate all address of device attached on cctalk network. Coin selector responds with only one byte (*non-standard message format*), after a delay that is proportional to address value multiplied with 4 milliseconds.

Message format is:

Host sends: [00] [00] [01] [FD] [Chk] Broadcast message

Coin s. responds: Dly -> [Address]

Example of message string for coin selector(address 2) is:

Host sends: [00] [00] [01] [FD] [02]

Coin s. responds: Dly=8 ms -> [02] Address is 2

Example of message string for coin selector (address 250) is:

Host sends: [00] [00] [01] [FD] [02]

Coin s. responds: Dly=1 s -> [FA] Address is 250

6.5.3.2 Command 252 [hexFC], Address clash

Command Address clash has same responds from coin selector but host issue this command with specific device address. Coin selector responds with only one byte (*nonstandard message format*), after a random value of time delay to prevent collision if two devices share same address. Message format is:

Host sends: [Dir] [00] [01] [FC] [Chk]

Coin s. responds: Random Dly -> [Address]

Example of message string for coin selector(address 2) **AL06V-c** is:

Host sends: [02] [00] [01] [FC] [01]

Coin s. responds: Random Dly -> [02] Address is 2

6.5.3.3 Command 251 [hexFB], Address change (not supported by coin acceptors)

Command Address change is issued to a specified device only. Coin selector responds with ACK message. Message format is:

Host sends: [Dir] [01] [01] [FB] [Address] [Chk]

Coin s. responds: [01] [00] [02] [00] [FD] ACK

Example of message string for coin selector(address 2) change to address to 20:

Host sends: [02] [01] [01] [FB] [14] [ED]

Coin s. responds: [01] [00] [02] [00] [FD] ACK Address is now 20

Coin selector does not responds to attempt of change an address to 0 or 1.

6.5.4 ALBERICI specific commands

First command header for specific factory setting and testing is **255 Factory set-up and test**. This command has several modes of use and some of them are factory secret and are not explained in this document.

Beside this next chapter describe two modified cctalk commands (Commands 214 Write data block and 215 Read data block) for encrypted exchange of data between host and coin selector.

6.5.4.1 Command 255, Factory set-up and test

This instruction header is used only for factory testing and programming!

With instruction command header one data byte must be sent for definition of command mode. Modes are:

- Mode 0(hex 00) No description – reserved for factory use only
- Mode 1(hex 01) No description – reserved for factory use only
- Mode 2(hex 02) Read coin selector memory data
- Mode 3(hex 03) Write coin selector memory data
- Mode 4(hex 04) Analog circuit test
- Mode 5(hex 05) Coin parameter test
- Mode 6(hex 06) Up-grade FLASH(program memory)
- Mode 7(hex 07) Re-initialization

First two modes reserved for factory use - not described in this document!

To use read and write instruction user must have basic knowledge about coin selectors memory data organization (Details are available on customer request, see document AL55/66-MemDataOrg-v1.pdf)

6.5.4.1.1 Command 255 mode 2, Read coin sel. memory

This instruction sends back to host block of memory data that was requested.

Data memory of coin selector is divided into a 6 group:

- Coin channel data address hex 0600 do 06FF(256 byte-a)
- Input/Output data address hex 0700 do 073F(64 byte-a)
- Factory common data address hex 0740 do 079F(96 byte-a)
- User general data address hex 07A0 do 07E9(74 byte-a)
- Statistic setting address hex 07EA do 07FD(20 byte-a)
- Statistic counters address hex 0800 do 083F(64 byte-a)

Factory key or user PIN protection are some time set to disable the access to some blocks of memory. Usually all memory is accessible for red instruction.

On any attempt to read memory location that is protected without clearing key or PIN protection mechanism, coin selector will respond with **NAK** message.

If number of blocks to read exceed block range, coin selector will also responds with **NAK** message. Maximum memory block read of 137 data bytes are limited by coin selector transmit buffer that has 142 bytes (Destination address+Bytes nr.+Source address+Header+59 data+Checksum=142). If block size extend this number coin selector will responds with NAK message to. Message string format is:

Host sends: [Dir] [04] [01] [FF][02][Start add-hi][Start add-lo][data nr.] [Chk]

Coin s. responds: [01] [n] [Dir] [00] [data 1] [data 2] . . . [data n] [Chk]

Start add-hi is start address hi byte, while start add-lo is address low byte. Data nr. represent the number of data to read (block size).

Data 1 to data n are requested coin selector memory data.

The example of message string for reading of first coin channel data is:

Host sends: [02] [04] [01] [FF][02] [06][00] [10] [E2]

Coin s. responds: [01][10][02][00][89][87][B4][77][A7][9F][08][08][0C][16][0A][04][08][01][09][C8][52]

6.5.4.1.3 Command 255 mode 4, Analog circuit test

This command returns to host string of data that can be used to test analog measuring circuit state. Details are described in internal documents and not available to users.

6.5.4.1.4 Command 255 mode 5, Coin parameter test

This command returns to host string of data from the last coin measurement.

This data are coin measured parameters and could be used to create and analyze coin data base.

Message string format is:

Host sends: [Dir] [01] [01] [FF][05] [Chk]

Coin s. responds: [01] [6] [Dir] [00] [data 1] [data 2] . . . [data 6] [Chk]

42 20 x 3 byte of memory!

43 Special case write command

Data string description:

- data 1 Measured coin parameter 1, AM1
- data 2 Measured coin parameter 2, PH1
- data 3 Measured coin parameter 3, AM2
- data 4 Measured coin parameter 4, PH2
- data 5 Measured coin parameter 5, PI3
- data 6 Measured coin parameter 6, DIM

The example of message string after insertion of one Euro coin for coin selector **AL66** (address 2) is:

Host sends: [02] [01] [01] [FF][05] [F8]

Coin s. responds: [01] [6] [02] [00] [8A] [80][8C] [5C] [AB] [8A] [D0]

Coin measured parameters(*data*) are available for read till next coin insertion.

This instruction is used by our coin programming software.

6.5.4.1.5 Command 255 mode 6, Up-grade FLASH

This command is used to up-grade coin selector FLASH program memory.

User can up-grade coin selector in cases when he knead some new function or improvement, which where not available at the moment when coin selector was purchased. This instruction is used by our coin selector programming software and generally is not useful to most users. Additional information are available on request.

Up-grade files are encrypted in factory and decryption is done internally by monitor program of coin selector. Coin selector will accept only correct type of up-grade file!

6.5.4.1.6 Command 255 mode 7, Factory reset

This command will set all coin selector data to initial factory programming values!

Warning: This could lead to unwanted coin selector function if different coin, input/output or user setting data where programmed!

The example of message string reset the factory settings is:

Host sends: [02] [01] [01] [FF][07] [F6]

Coin s. responds: [01][00][02][00] [FD] ACK

After that, coin selector must be switched off/on!

6.6 Error Codes

PLEASE NOTE: default status does not include Error Codes. The coin acceptor can be preset for error code transmission IF REQUESTED WHEN PLACING THE ORDER FOR THE COIN ACCEPTOR(S).

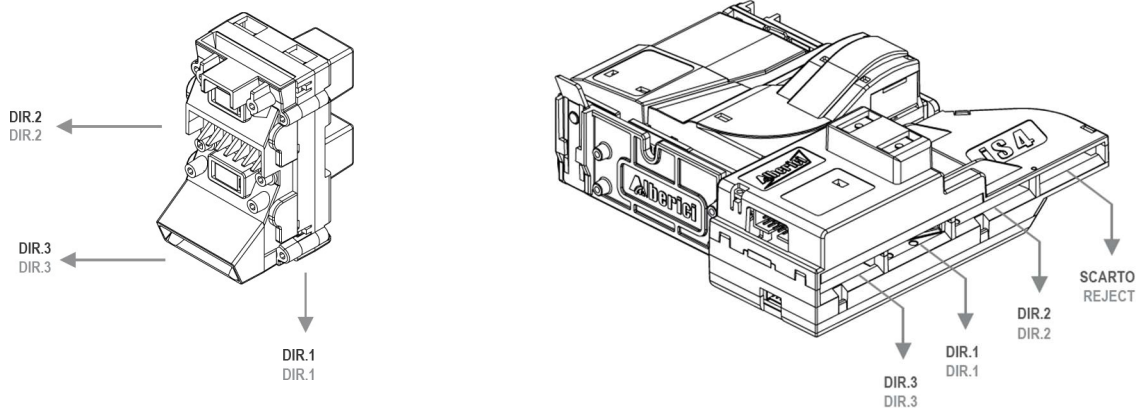
ERROR CODE LIST

| Code d. | Code h. | Error | Coin rejected |
|---------|---------|--|---------------|
| 0 | 00 | Null event | No |
| 1 | 01 | Reject coin (not recognized) | Yes |
| 2 | 02 | Inhibited coin (master inhibit) | Yes |
| 3 | 03 | Multiple window (fraud or similar coin) | Yes |
| 5 | 05 | Validation (measuring) time out | Yes |
| 6 | 06 | Credit sensor (recognition to opto 2) time out | Possible |
| 8 | 08 | Second close coin | Yes/both |
| 11 | 0B | Sorter not ready | Yes |
| 13 | 0D | Validation sensor not ready | Yes |
| 16 | 10 | Credit sequence error (Jojo) | No |
| 18 | 12 | Coin to fast (opto 2 minimum time not elapsed) | No |
| 19 | 13 | Coin to slow (opto 2 time out) | No |
| 128 | 80 | Inhibited coin (position 1) | Yes |
| ... | ... | Inhibited coin (position n) | Yes |
| 143 | 8F | Inhibited coin (position 16) | Yes |
| 255 | FF | Unspecified alarm code | - |

Else, make use of the Alberici Coin Selector programming software to enable transmission. Tick the checkbox in the menu "Config. CCTalk", "Settings"/"Activate Error Events in Coins Test", then press "Write data" key to transfer the instruction into the coin acceptor.

7. Driving SPI peripheral devices

To attribute the appropriate sorting commands to the iS1 and iS4 SPI SORTERS, please refer to the output paths shown below, and to the directions given in the section “SPI DISPLAY AD01 SERIAL PROTOCOL”.



| 8 PIN SPI (X2) | | | |
|----------------|---|---|---------|
| DATA OUT | 1 | 2 | CLOCK |
| SELECT | 3 | 4 | DATA IN |
| GND LOGIC | 5 | 6 | +5 Vdc |
| GND | 7 | 8 | +24 Vdc |

| 10 PIN PULSE (X1) | | | |
|-------------------|---|----|-------------|
| GND | 1 | 2 | +24 Vdc |
| OUT5 | 3 | 4 | OUT6 |
| IN2/OUT7 | 5 | 6 | IN1/INHIBIT |
| OUT1 | 7 | 8 | OUT2 |
| OUT3 | 9 | 10 | OUT4 |

SPI Display AD01: serial protocol

The Alberici display AD01 consists of standard 16x2 characters LCD module with parallel interface, background light, equipped with a small controller pcb for SPI communication with the host. It has been designed to be used as a display accessory for AL66 FG coin selectors, but it can be used with any other host capable to interface through SPI.

7.1 Communication specifications

Slave mode operation

8 bit rate

Maximum Clk freq. = 2 MHz

Clk phase = 0; Transfer on the first clock edge

Clk polarity = 0; Transfer on clock transition low to high

7.1.1 Clock frequency

Clock frequency is not critical but if it is too low the display will blink when new data or refresh of the old data is done. Maximum frequency is limited with connection cable capacity and should be max 2 MHz for short connection (less than 300 mm). Recommended frequency is above 100 KHz till 1 MHz.

7.1.2 Power supply

The background light of the display must be powered by 8-26 Vdc. The logic circuit must be powered by 5 Vdc. See connector in section 7.1.3.

7.1.3 Connection (SPI Interface)

| | | | |
|-----------|---|---|------------|
| DATA OUT | 1 | 2 | CLOCK |
| SELECT | 3 | 4 | DATA IN |
| GND LOGIC | 5 | 6 | +5 Vdc |
| GND | 7 | 8 | +12/24 Vdc |

The connector is pin-to-pin compatible with the one on the AL66 FG coin selectors

7.2 Message structure

The display can work only as slave and does not reply to any of the command messages sent by the host.

There are two different sets of commands:

- Display coin selector data
- Display message

First set is used to show specific coin selector data and second is used to show any display message.

7.2.1 Display coin selector data

The coin selector will send to the display controller the following command string:

[**0xC0**]

[**0x08**]

[**Display mode**]

[**Display multiplier and decimal position**]

[**Display light int.**]

[**Accumulated credit**]

[**Accumulated value**]

[**Time in minutes**]

[**Time in sec.**]

[**Display warnings**]

The first byte **0xC0** in message string means the type of message. The Hex code **0xC0** displays specific data of the coin selector.

The second byte is always **0x08** and represents the 8 bytes of data in the string.

Display mode is the code for showing further and different coin selector data. It is not possible to present all the data of the coin selector at the same time. Possible combinations and codes are shown in table below:

| Code | Function |
|------|-----------------------------|
| 0x00 | stop display |
| 0x01 | display Credit |
| 0x02 | display Value |
| 0x03 | display Credit and Value |
| 0x04 | display Time |
| 0x05 | display Credit and Time |
| 0x06 | display Value and Time |
| 0x08 | display Warning messages |
| 0x09 | display Credit and Warnings |
| 0x0A | display Value and Warnings |
| 0x0C | display Time and Warnings |

Table 2 AD01 - Coin selector display modes

A special case of message is when this byte is **0x00** and there are no other of data to consider. It is used to turn off the background light of the display, stop the operation of the display device, and set to power saving mode (*stop mode*) the MCU communication MCU. Details concerning such data are not described in this document.

The **Display multiplier and decimal position** code must be used when the inserted value must be displayed. Coin selectors use relative coin value for their programming - but this is not at all a practical information to be displayed for Users. Because of that, this value must usually be multiplied and decimals must be set.

For instance: if the 50 cent coin has been programmed with relative value = 1, in order to display its value as an understandable 0,50 (€), the relevant value 1 must be multiplied by 50. For this reason, 2 decimal positions must be made available. Tables 3a and 3b show the codes corresponding to the various multipliers and decimal positions.

| Code hex | Dec position |
|----------|-------------------|
| X0 | No decimal |
| X1 | One dec. place |
| X2 | Two dec. place |
| X3 | Three dec. places |

Table 3b Decimal places codes

| Code hex | Multiplier |
|----------|------------|
| 0X | 1 |
| 1X | 1 |
| 2X | 2 |
| 3X | 4 |
| 4X | 5 |
| 5X | 10 |
| 6X | 20 |
| 7X | 25 |
| 8X | 50 |
| 9X | 100 |
| AX | 200 |
| BX | 250 |
| CX | 500 |
| DX | 1000 |
| EX | 1000 |
| FX | 1000 |

Table 3a Value multiplier codes

The byte of the **Display light intensity** allows to set the duty cycle of the driver of the light brightness (*Pulse Width Modulation*).

When the display is connected to the coin selector, the intensity of its background light will then be stored into the coin selector NV memory (*FLASH*), and it will remain stable even if power supply is changed.

Warning: when the host sends the brightness value to the display, be aware that high power supply can damage the light driver or the background LED-s if a high brightness value has been set.

The **Accumulated credit** byte represents the amount of the accumulated credit for the coin selectors that are programmed for releasing the credit signal only after the "Request" command has been activated.

The **Accumulated value** can be the coin value inserted in the coin selector, or the credit value remaining after the credit impulse has been activated, or the remaining time after the timer count-down has been activated. The accumulated value will be displayed according to the multiplier and decimal position code!

The **Time** is displayed in format: 'mm : ss'. The coin selector will refresh the display every second as long as the timer output is active.

The **Display warnings** can show the status of the coin selector or other warnings. Warnings and relevant codes are shown in table 4.

| Code hex | Warnings |
|----------|----------------|
| 00 | Ready |
| 01 | Hardware error |
| 06 | Coin error |
| 10 | Rest value |
| 20 | Credits |
| 40 | Time out |

Table 4 Warning codes

Any other code sent will produce no result on the display, so this one will be blank.

7.3 Display message

The Host can send commands to show any text on a 16x2 digits display. The following command strings must be used:

```
[ 0xC1 ]
[ number of bytes ] [ Character address ] [ Data 1 ]
[ Data 2 ]
.....
[ Data n ]
```

First byte **0xC1** string means the type of message. The Hex code 0xC1 is for general message.

The second byte means the amount of **data bytes (n)** in the message, including the address (n+1).

The third byte corresponds to the character address: it starts by 0x00 for the beginning of the first digit on first line (top), and finishes by 0x1F for the last digit on second line (bottom). The

second line begins by address 0x10.

The bytes that follow share the same structure of the character codes of the Display table (Table 4).

8. Maintenance

To ensure perfect functioning, clean the coin path approx. every 10.000 dropped coins.
Take special care of the dirt on the optical sensors.

1 Cleaning the coin path

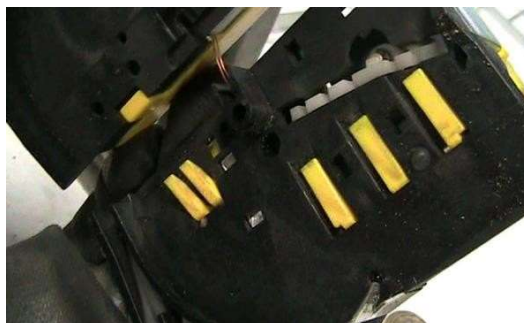
Use a clean cloth moistened with glass cleaning product. Do not use smeary liquids.



2 Cleaning the optical sensors:

Insert a flat screwdriver as in the picture, then carefully force the lower cover up and out.

Pay attention not to break the plastics, and take care not to stretch or break the copper wires of the electric coil.



Remove dust and dirty particles from the optical sensor prisms.
If matt or scratched, the prisms must be replaced.



9. Disposal of the product



WARNING! DISPOSE OF ACCORDING TO THE GOVERNING LAW IN YOUR COUNTRY!
This equipment may not be treated as household waste. Instead, it must be handed over to the applicable collection point for the recycling of electric and electronic equipment. By ensuring that this product is disposed of correctly, you will help to prevent potential negative consequences for the environment and human health, which could otherwise be caused by inappropriate waste handling of this product.

For more detailed information about recycling of this product, please contact the Dealer where you purchased this product.

10. Warranty

The manufacturer will fix malfunctions arising from production faults in this machine or parts of it within 12 months from the date of sale.

All communications referring to guarantee repairs or replacements must be accompanied by the product serial number and the copy of the sale invoice.

To obtain your guarantee repair, please send the part to the Dealer where you purchased the machine, together with the following documents:

- copy of the sale invoice
- delivery note stating "returned for guarantee repair"
- detailed report of the problem found and the circumstances in which it occurs.

Before sending the product, please get in touch with your Dealer +43 (0) 1 271 66 00 very malfunctions can be fixed via a simple phone call, saving you costs and time.

Alberici S.p.a. will verify that warranty is applicable, i.e. that problem is not caused by:

- transport damages
- damages from incorrect installation or wrong configuration
- installation in premises or areas not complying with the prescribed safety requirements
- intentional or unwilling tampering
- wrong or careless use or maintenance
- non-compliance with precautions prescribed (see Chapter 4. Caution)
- natural disasters, vandalisms, intentional or unintentional damage

Guarantee is considered automatically expired if outer and inner labels are missing.

Transport costs of repaired products are at the Customer's charge.

11. Customer Service

A.u.S. will be pleased to offer all the necessary information on use, ordinary maintenance and technical service. Please call (+43) (0) 1 271 66 00 and specify if your request concerns information on use or technical support.

COMPLIANCE DECLARATION



DIRECTIVE 2014/35/CE - DIRECTIVE 2014/30/UE

La ditta **Alberici S.p.A.**, avente sede in via **Ca' Bianca, 421, 40024 Castel San Pietro Terme (BO) – Italia**,
D I C H I A R A

Che il sistema classificato nella famiglia di prodotto **apparecchio elettrico d'uso domestico e similare – dispositivo elettronico: gettoniera**, finito di costruire ed assemblare il / / , identificato univocamente da:

| Modello | Configurazione | Protocollo | N° di Serie e/o matricola |
|----------------|---|---|---------------------------|
| AL66 FG | <input type="checkbox"/> V <input type="checkbox"/> K | <input type="checkbox"/> Aes256 DH <input type="checkbox"/> ccTalk | ----- |
| | <input type="checkbox"/> S <input type="checkbox"/> I | <input type="checkbox"/> Aes2048 DH <input type="checkbox"/> Standard | |

Essendo realizzato conformemente al modello prototipo campione denominato AL66 FG V, finito di testare positivamente ai fini EMC e LVD (rapporto 7037CE-AL66.doc) il 25/09/2015, dalla STP S.r.l., con sede legale in via P.F. Andrelini, 42, 47121 Forlì (FC), Italia e sede operativa in via San Donnino, 4, 40127 Bologna (BO), Italia, risulta essere conforme a quanto previsto dalle seguenti direttive comunitarie:

- a) le norme armonizzate (per i punti applicabili):
- | | |
|---|---|
| <ul style="list-style-type: none"> - CEI EN 55014-1 (CEI 110-1); - CEI EN 55014-2 (CEI 210-47); - CEI EN 55022 (CEI 110-5); - CEI EN 55024 (CEI 210-49); - CEI EN 60065 (CEI 92-1); - CEI EN 60335-1 (CEI 61-150); - CEI EN 60335-2-82 (CEI 61-226); - CEI EN 60950-1 (CEI 74-2); - CEI EN 61000-3-2 (CEI 110-31); | <ul style="list-style-type: none"> CEI EN 61000-3-3 (CEI 110-28); CEI EN 61000-4-2 (CEI 210-34); CEI EN 61000-4-3 (CEI 210-39); CEI EN 61000-4-4 (CEI 210-35); CEI EN 61000-4-5 (CEI 110-30); CEI EN 61000-4-11 (CEI 110-29); CEI EN 61000-6-1 (CEI 210-64); CEI EN 62233 (CEI 61-251). |
|---|---|
- b) In conformità ai requisiti essenziali di sicurezza della Direttiva Bassa Tensione:
- L. 791 del 18 Ottobre 1977 e s.m.
 - 2014/35/UE del 26 Febbraio 2014;
- c) in conformità ai requisiti essenziali di sicurezza della Direttiva Compatibilità Elettromagnetica:
- D.Lgs. 194 del 06 Novembre 2007.
 - 2014/30/UE del 26 Febbraio 2014;

Che conferiscono la presunzione di conformità alla Direttiva 2014/30/UE

Castel San Pietro Terme (BO), Italia li, / /

Fabrizio Alberici
 Il Presidente

NOTICE

Alberici S.p.A. reserves the right to make changes to the equipment described and to its technical specifications at any time and without notice, in pursuit of continual improvement of this product.



Via Ca' Bianca 421
40024 Castel San Pietro
Terme (BO) – ITALY

Progettazione e produzione di sistemi di pagamento, accessori per videogames e macchine vending
Design and manufacture of payment systems, accessories for videogames and vending machines

Tel. + 39 051 944 300
Fax. + 39 051 944 594

<http://www.alberici.net>

info@alberici.net